

Hybrid Algorithms of Laser Scanning Point Cloud for Topological Analysis



Vladimir Badenko , Alexander Fedotov  and Konstantin Vinogradov 

Abstract Laser scanning technologies are widely used to solve civil engineering problems and land use management in a GIS environment including digital terrain models (DTMs) creation. Some gaps in raw laser scanning data processing algorithms for DTM are analyzed. Algorithms for filtration, triangulation, and defragmentation are proposed. Advantages and disadvantages of the algorithms proposed are discussed. Triangulation algorithm can serve to defragment cloud of laser scanning points into semantic component parts. Defragmentation includes recognition of engineering objects and other objects of the terrain and their delineation. Results of applications to real problems show the robustness of algorithms proposed.

Keywords Algorithm · Laser scanning · Cloud of points · Triangulation
Digital terrain model

1 Introduction

One of the developing and most contemporary technologies of the active remote sensing is laser scanning. Nowadays, the results of laser scanning are a base for creation of three-dimensional (3D) models of different objects. Appropriate technologies based on 3D data have been using all over the world in different applications: road management, urban modeling, industry survey, heritage conservation [1]. The types of remote sensing laser scanning systems are the following: terrestrial, mobile,

V. Badenko (✉) · A. Fedotov
Peter the Great St. Petersburg Polytechnic University,
Polytechnicheskaya 29, 195251 St. Petersburg, Russia
e-mail: vbadenko@gmail.com

A. Fedotov
e-mail: afedotov@spbstu.ru

K. Vinogradov
Saint-Petersburg State University, Universitetskaya Emb. 13B, 199034 St. Petersburg, Russia
e-mail: kostya1495@mail.ru

and airborne [2]. During an object surveying, the distance from sensor to object is determined on the base of time delay (pulse method) or the phase shift (phase method) of laser beam reflected by object surface [3]. A cloud of points is the result of laser scanning survey. Obtained laser scanning point clouds are usually combined with digital photos [4]. As a result, for each point in such cloud the following set of information about object surface is stored: three coordinates, intensity of signal reflected and R/G/B color. As it can be seen that information describes the surveyed object quite well for modeling. It should also be noted that in recent years, there has been a significant increase in the geometrical accuracy and density of points in laser scanning clouds [5].

Laser scanning technologies are widely used to solve civil engineering problems and land use management in a GIS environment [6–9]. There are also many publications on the successful use laser scanning cloud of points for digital terrain models (DTMs) creation [10–12]. One of the most popular models for DTM is Delaunay triangulation, which provides the most qualitative, uniquely defined, and invariant, in relation to the coordinate system, the construction of the model (triangulation) [13]. At the same time, the usage of Delaunay triangulation is associated with a large expenditure of time and computer memory, which is a matter of actual importance for constructing a model from the big cloud of laser scanning points [14]. The task of DTM creation is one of the most urgent tasks of processing of laser scanning results despite the long history of the development of appropriate algorithms [15–19].

The objectives of this paper are to analyze some gaps in algorithms for processing of raw laser scanning data for DTM creation. Some new algorithms which can bridge the gaps are presented. The algorithms proposed include combination of different approaches, and this is why they are named “hybrid.” A triangulation algorithm can serve to defragment cloud of laser scanning points into semantic component parts. Defragmentation includes recognition of engineering objects and other objects of the terrain and their delineation. The results presented in the paper can serve as a basis for solving many civil engineering problems: not only such as creation of a DTM, but also can be useful for processing laser scanning data during engineering survey of roads and existing buildings.

2 Materials and Methods

2.1 Filtration of Points Density

A point will be treated as a data structure ($a \cdot x$, $a \cdot y$, and $a \cdot z$) that contains spatial coordinates and attribute information: color, time, intensity of the reflected laser measurement signal, etc. The lexicographic order of sorting points defines following relations between points a and b : $a > b$, $a < b$, and $a = b$. The lexicographic sort order is specified by a function $F(a, b)$, which returns the result of comparing two elements (points). The function returns the following values: $F(a, b) = -1$ if $a < b$; $F(a, b) = 1$

if $a > b$; $F(a, b) = 0$ if $a = b$. For the triangulation algorithm, various variants of the lexicographic comparison “ $x-y-z$ ” or “ $y-x-z$ ” are used depending on the direction of “elongation” of a cloud of laser scanning points under discussion. Nevertheless, further only one order of lexicographic comparison will be considered: “ $x-y-z$.”

A filtration problem $OUT = F(IN)$ is considered. From the original set, IN of points is needed to select such a subset of points $OUT \in IN$, that for all points of from OUT the following two conditions are fulfilled:

1. Conditions in the plane (for coordinates x and y):

$$|a \cdot x - b \cdot x| > eps \text{ and } |a \cdot y - b \cdot y| > eps, a \in OUT, b \in OUT \text{ and } OUT \in IN \quad (1)$$

Here, eps is a parameter that determines a value of the desired density of points. Such filtering ensures that there are no points in the OUT subset; the distance between them is less than the value of eps . It is possible also to set the condition in the form:

$$\sqrt{(a \cdot x - b \cdot x)^2 + (a \cdot y - b \cdot y)^2} > eps, \quad (a \cdot x - b \cdot x)^2 + (a \cdot y - b \cdot y)^2 > eps^2 \quad (2)$$

But for its implementation requires large computational costs without a tangible gain in quality.

2. Condition for height (for coordinate z).

From the subset Q , for any points of which the identity condition is satisfied:

$$|a \cdot x - b \cdot x| < eps \text{ and } |a \cdot y - b \cdot y| < eps, a \in Q, b \in Q \text{ and } Q \in IN, \quad (3)$$

only one point is chosen $\min(a \cdot z)$ with minimal z value.

The solution of the filtration problem in this formulation is used primarily to ensure stability of a triangulation algorithm proposed. Filtration algorithm is used to form a surface using a cloud of laser scanning points. It can be used to reduce the number of points. The filtration algorithm is implemented in two stages:

1. A set of laser scanning points is sorted by values of z coordinates.
2. All points are being included in a sorting container, which contains only unique values. The container can be a search binary tree or an associative container *set* from STL (Standard C++ Template Library).

For sorting at this stage, a following function is defined:

```
//Example of the lexicographic comparison function for
filtering in plane (x, y)
static float heps = eps / 2; // half
int filterPointXY(Point a, Point b)
{
    if (a.x < (b.x - heps)) return -1;
    if (a.x > (b.x + heps)) return 1;
    if (a.y < (b.y - heps)) return -1;
    if (a.y > (b.y + heps)) return 1;
    return 0;
}
```

It is not difficult to recognize that for such comparison function, the points a and b are considered to be coincident, if following condition is true:

$$|a \cdot x - b \cdot x| < eps \text{ and } |a \cdot y - b \cdot y| < eps. \quad (4)$$

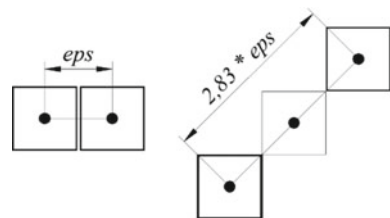
The second stage ignores all non-unique (identical) points, except the first. Since at the first stage, the z sorting has already been performed and the lowest points are the first in the container, it can be ensured that after the second stage the necessary subset of the bottom points has been created with the necessary lexicographical order. The actual range of filtration density d is within the following limits (Fig. 1):

$$eps < d \leq 2 * \sqrt{2} * eps \quad (5)$$

An alternative filtering variant was considered without performing the first stage. When a point is moving to a container, if there were “identical” points in the plane (x, y) , then a point with smaller coordinate z is chosen (Fig. 2).

This approach forms a smaller subset of filtered points than a subset of the first variant. This can be explained by the fact that the lowest point does not the first in the container, and the previous point-candidate for a lowest value has already supplanted some of the point-candidate (migration of “bottom” points and filtration zones). Figure 2a shows the point and the filtering zone of size eps . If, as a result of filtering, the minimum height is overridden, the filtering zone, which size can significantly exceed the original one, is migrated (Fig. 2b). Conflicts between filtering zones may also arise, which will result in the need to remove one of the conflicting zones (Fig. 2c). In general, for this algorithm, only the lower limit of the density

Fig. 1 Actual range of filtration density



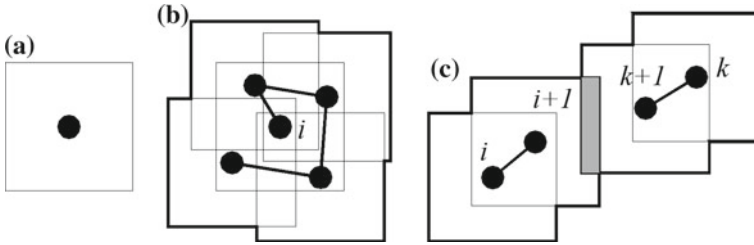


Fig. 2 Migration of the filtering area

range eps can be guaranteed, and the upper one is increased and cannot be reliably estimated.

The effectiveness of the first variant is that the preliminary sorting by z is done once for original point cloud, and in general, the filtration can be performed several times only on the second stage of filtration for different values of eps . It should be noted that this approach is easy to implement within the binary search tree, associative container *set* or for hash tables. In the case of hashing, the sorting order (for example, x - y - z) determines the order in which the composite index is generated for the hash function. The high sorting efficiency (in our case of filtering) within the binary search tree is determined by the fact that the search costs are determined by the function $O(\log n)$, in contrast to the sequential search $O(n)$. Here n is the number of points in the original set.

2.2 Triangulation

Hybrid algorithms proposed here should be considered as a result of a compromise between the performance of the triangulation and the quality of the model. As initial data for triangulation, a set of filtered points sorted in the plane (x, y) , according to the proposed lexicographic order of sorting is considered. The process of triangulation involves the formation, at the initial stage, of a first face (triangle) of first 3 points, and then successively attaching remaining points. In fact, it is assumed that there is a real situation where first m points in the plane (x, y) lie on one line (and maybe all points), then the algorithm is terminated.

Let a point with an index $m+1$ does not lie on the line of the first m points, so $m - 1$ triangles can be formed on a base of this point.

For any step of the triangulation, the following two statements are hold:

1. Outer edges of the triangulation form a convex boundary. This statement is obvious, since the starting set of triangles has a convex boundary. Any joining of a new point is also performed until the formation of a new convex boundary. At the same time, a new set of triangles joins to the triangulation (set *begin-end-i* on Fig. 3).

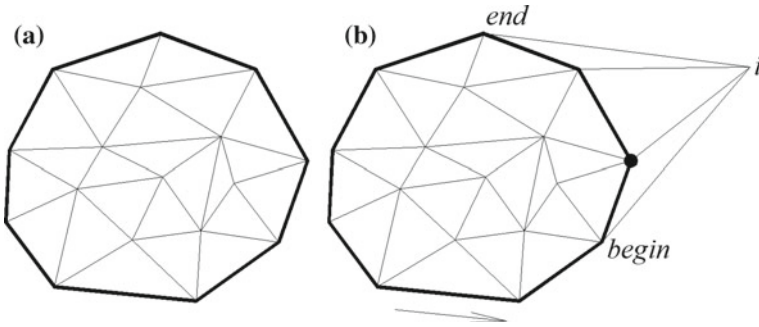
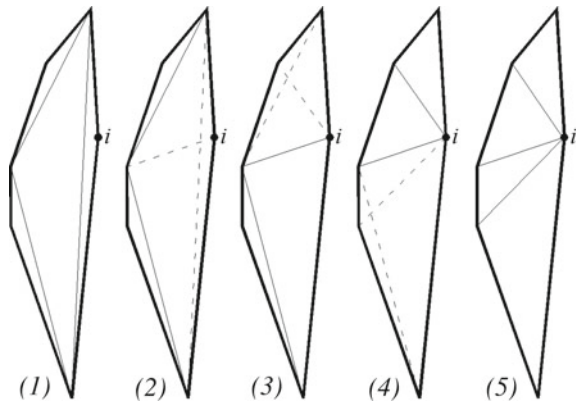


Fig. 3 One step of the triangulation algorithm

Fig. 4 Sequence of optimization steps: The dashed lines indicate the test edges to be compared



2. Any joining point lies outside the convex boundary of the triangulation. This condition is ensured as a result of the specified lexicographic sort order at the point filtration stage.

After each step of joining the point, optimization for the set of triangles begin-end-*i* is needed. Optimization is based on comparing two adjacent edges, where the length of their common edge is compared with the distance between unbound vertices (Fig. 4). An attempt to connect these unconnected vertices has been made. If (1) the common edge and the test edge between the vertices intersect and, (2) the length of the test edge is shorter than the common edge, then a general quadrangle is rebuilding. The common edge is removed and a new edge replaces the test edge. It should be noted that the optimization is not performed to the full depth and is limited to the edges incident to the vertex *i*. This serves to limit the optimization process on the principle of “minimal diagonal” and allows creating a surface model with a sufficient quality.

So, the triangulation algorithm consists of two main steps:

1. Forming the starting set of triangles.
2. A cycle on a set of laser scanning points (excluding starting points):

- 2.1. For a i th point and the convex boundary, the search for the beginning and end (*begin*, *end* on Fig. 3) of the new set of triangles is performed. The search begins at $i - 1$ point, since it is lexicographically (but not geometrically) the closest point to i .
- 2.2. Joining the triangles of the set of triangles *begin-end-i*.
- 2.3. Optimization of triangles of the set of triangles *begin-end-i*.
- 2.4. Change of convex boundary. Now it passes through the vertices *begin-end-i*.

3 Results and Discussion

3.1 Discussion of the Filtration Algorithm

Using this filtering algorithm for direct “land detection” based on the results of airborne laser scanning of the terrain has been analyzed. In this case study, approximations of DTMs are strongly offset in the direction of removal (cutting) of protruding (elevated) fragments of the relief microforms.

Another difficulty is connected with the fact that in the presence of civil engineering structures, and other objects that are not a part of the terrain. It is necessary to choose the value of *eps* exceeding their geometric dimensions. Only in this case, it can be assumed that the filtered set contains the relief points.

Some advantages of this filtering algorithm can be noted:

1. The lower (and upper) boundary of a filtering range is guaranteed, which allows taking into account and controlling the numerical stability of the triangulation algorithm.
2. From a set of measured points, directly measured points are selected. This is very important for the future solution of the problem of revealing structural relief lines, on the basis of its morphostructural analysis. This will allow performing composition (or merging) of meshes with different density without damaging the introduction of distortions in the calculations due to the filtering method used.
3. The method used has high productivity and flexibility. This is especially true when processing large clouds of laser scanning points.

3.2 Discussion of the Triangulation Algorithm

Features of the triangulation algorithm proposed are following:

1. All search operations are performed on the outer boundary of the triangulation and do not depend on the number of its vertices and edges. Search is not performed across the entire border, but to the right and left of the lexicographically nearest point (the previous point). Even with optimization, adjacent faces are determined

not from the search, but from an editable list of edges relationships. These search features determine high processing speed and linear cost growth with increasing number of source points.

2. Formation of triangulation develops in the positive direction of the X -axis (for sorting x - y - z , and for sorting by y - x - z —along the Y -axis) and over the entire width of the front. This makes it easy to split the model into fragments of a given size in the direction of triangulation. It should be noted that the implementation of the algorithm itself does not depend on the order of sorting x - y or y - x . The sorting order only affects a triangulation direction. If the initial set of points has an oblong directionality, then the order of sorting is selected (according the ratio of the sides of the bounding rectangle). Managing sort order allows you to optimize the costs of implementing calculations.
3. Stability of the algorithm depends on the parameter, which is determined by the ratio of the density parameter to the width of the front of the triangulation $k = eps/R$. Here, R is the maximum possible distance between two lexicographically closest points. This value does not exceed the width of the front of the triangulation at its widest point. If the sort order x - y - z is used, then the triangulation develops in the positive direction of the X -axis and the triangulation front is the maximum width along the Y -axis. For a given value of eps , it is possible to estimate the maximum possible width of the triangulation front and predict the numerical stability of the computational process and also recommend the sizes of the processed fragments.

It should be noted some disadvantages of the algorithm proposed (in comparison with the Delaunay triangulation):

1. The algorithm is conditionally optimal in part of the geometry of triangles (faces). Optimization is performed at a limited depth. At the same time, a number of non-optimal faces on test calculations do not exceed 1–2%.
2. The algorithm is not invariant with respect to a coordinate system and a triangulation direction (sorting direction). When a triangulation direction is changed, new version of the surface model is created. It should be noted that the magnitude of these discrepancies is insignificant and fits into the number of non-optimal faces.

Despite the listed disadvantages, the advantages of this kind of triangulation for processing large data sets do not cause any doubts.

At this stage, the differences in quality between the Delaunay triangulation and the proposed approach do not have a significant effect on the process and the quality of the defragmentation (due to the high density of points). But productivity, in turn, differs by almost an order of magnitude. In the subsequent stages, during formation of final DTM, the Delaunay triangulation with constraints in the form of structural lines is used. But here the density of the points and the size of the modeled fragments have been substantially reduced.

Comparison of dependence of calculation time of the triangulation versus the number of initial points shows that the growth in costs in the Delaunay triangulation is determined by an almost quadratic dependence and in proposed case almost linear dependence.

3.3 Application of the Algorithms Proposed

The proposed algorithm of triangulation of a laser scanning points cloud leads to formation of a convex boundary. This does not always correspond to the real situation. However, inclusion in the algorithm possibilities of removing edges of the outer border, which exceeds the length allowed by the user, allows solving this problem. Figure 5 shows an example of application of such algorithm modification. On the left is a triangulation without delineation with a convex outer boundary. On the right is the result of contouring with a non-convex outer boundary.

The method based on the use of triangulation models allows analyzing the connections of the nearest points, dividing the model into fragments and performing their topological analysis, visualizing the intermediate results in the form of surfaces. At the first stage, a surface is formed, which, as a result of transformations, smoothing, and other possible procedures, is brought to a form that the user can recognize as the best approximation to the terrain. The proposed approach is based on such concepts as the boundary between adjacent faces and the “sloping” and “vertical” parts of the faces. To determine the characteristics of the face—“sloping” or “vertical,” a comparison of the vertical excess and the maximum length of a facet edge projected onto a plane (or on the plane of an adjacent face) is used.

This approach allows implementing various strategies for defragmenting models. Depending on belonging to the external or internal border, it is possible to uniquely determine the hierarchy of relations between regions (fragments). Internal areas are also subject to defragmentation, and the results of dependencies between areas complement the hierarchy tree of objects. Such a selection of the region is performed for all faces of the original surface, if they did not fall into any of the regions in the previous stage of defragmentation. Thus, entire investigated object is divided into regions (fragments) and a hierarchy tree of their relations (by nesting) is formed, which in turn serves as a basis for topological analysis of models (Fig. 6).

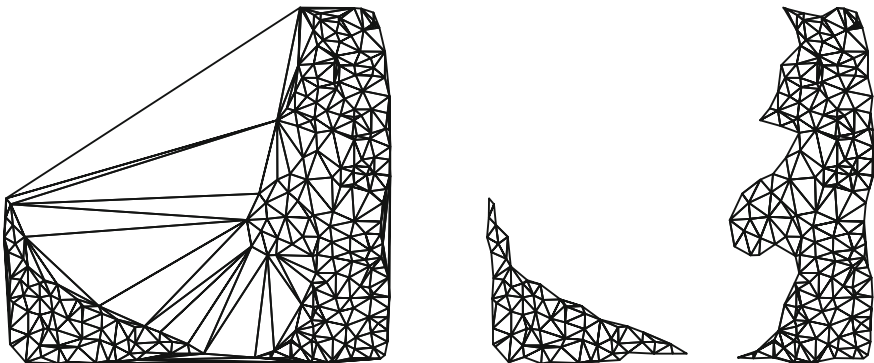


Fig. 5 Contouring of the convex boundary of a triangulation

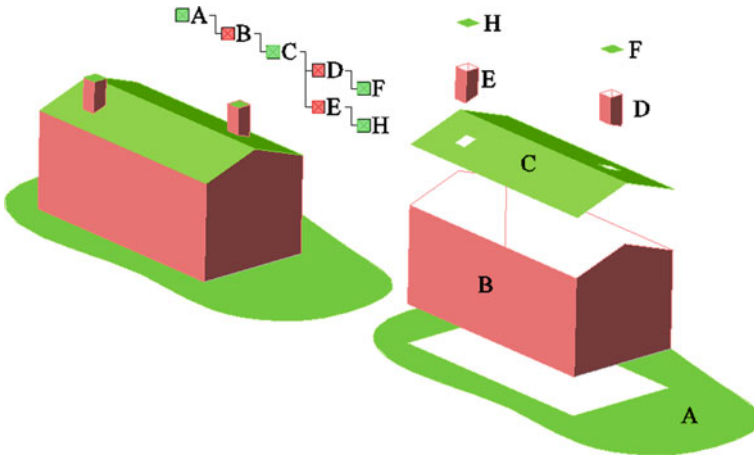


Fig. 6 Example of topological defragmentation of a civil structure

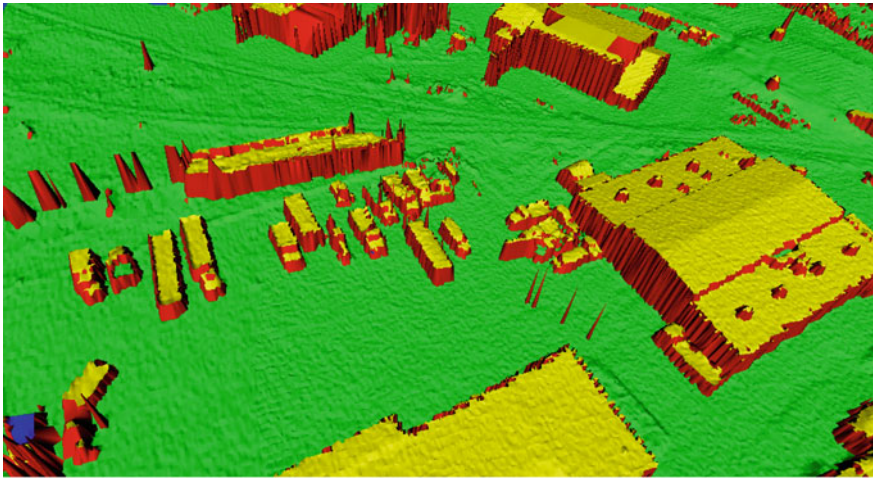


Fig. 7 Automatic classification of objects of the “roof” type (yellow)

For each fragment, an array of additional statistical information is also formed. For example, for each external boundary of the region, a number of neighbor fragments is determined, as well as how many of them are pointing upwards (excesses are positive), and separately a number of faces directed downwards (excesses are negative). In general, based on such statistical data, as a result of topological analysis, it is possible to classify fragments by types: “earth,” “roof,” “wall.” In Fig. 7, results of classify objects of the “roof” type are presented. This approach does not guarantee the absence of errors, but gives satisfactory results.

4 Conclusions

It is planned to use various options for filtering points to highlight structural relief lines. Within the framework of the morphostructural analysis of the relief, there are algorithmic solutions for the allocation of structural relief lines over several DTMs constructed with different density of points (ref). The basic idea of such algorithms is that it is possible to construct a difference of two surfaces. For example, the first approximation of the surface relief with a density of points at $eps = 40$ meters and the second surface is more dense—at $eps = 10$ meters. Variations in the deviations of this difference surface from the plane (e.g., in the distribution of the volume of the earthen mass, calculated by the method of trihedral prisms) determine the direction in which the position of the structural line can be refined purposefully or iteratively.

The results presented in the paper can serve as a basis for solving many civil engineering problems: not only such as creation of a DTM. The presented methods can be used for processing laser scanning point clouds during engineering survey of roads and existing building. Automatically create a quality DTM using raw laser scanning data in form of a point cloud is almost impossible. The proposed algorithms are another example of numerous techniques for refining and mapping the terrain. Nevertheless, the proposed approaches to DTM creation minimize needs for editing and thus increase the productivity of out-of-field works.

Acknowledgements The research was supported by Ministry of Education and Science of Russia within the framework of the Federal Program “Research and Development in Priority Areas for the Development of the Russian the Science and Technology Complex for 2014–2020” (project ID RFMEFI58417X0025).

References

1. Guan H, Li J, Cao S, Yu Y (2016) Use of mobile LiDAR in road information inventory: a review. *Int J Image Data Fusion* 7(3):219–242
2. Heo J, Jeong S, Park H-K, Jung J, Han S, Hong S, Sohn H-G (2013) Productive high-complexity 3D city modeling with point clouds collected from terrestrial LiDAR. *Comput Environ Urban Syst* 41:26–38
3. Wang C, Cho YK (2015) Smart scanning and near real-time 3D surface modeling of dynamic construction equipment from a point cloud. *Autom Constr* 49:239–249
4. Tang P, Huber D, Akinci B, Lipman R, Lytle A (2010) Automatic reconstruction of as-built building information models from laser-scanned point clouds: a review of related techniques. *Autom Constr* 19(7):829–843
5. Yang B, Dong Z (2013) A shape-based segmentation method for mobile laser scanning point clouds. *ISPRS J Photogrammetry Remote Sens* 81:19–30
6. Shivpuje P, Deshmukh N, Bhalchandra P, Khamitkar S, Lokhande S, Jondhale V, Bahuguna V (2016) Investigation for land use and land cover change detection using GIS. In: *Proceedings of the international congress on information and communication technology*. Springer, Singapore, pp 393–399
7. Fedorov M, Badenko V, Maslikov V, Chusov A (2016) Site selection for flood detention basins with minimum environmental impact. *Procedia Eng* 165:1629–1636

8. Barazzetti L (2016) Parametric as-built model generation of complex shapes from point clouds. *Adv Eng Inform* 30(3):298–311
9. Badenko V, Kurtener D, Yakushev V, Torbert A, Badenko G (2016) Evaluation of current state of agricultural land using problem-oriented fuzzy indicators in GIS environment. In: LNCS, vol 9788. Springer, Heidelberg, pp 57–69
10. Bater CW, Coops NC (2009) Evaluating error associated with lidar-derived DEM interpolation. *Comput Geosci* 35(2):289–300
11. Höhle J, Höhle M (2009) Accuracy assessment of digital elevation models by means of robust statistical methods. *ISPRS J Photogrammetry Remote Sens* 64(4):398–406
12. Li H, Wang X, Zhu J, Li W (2014) A method of airborne LiDAR data classification based on curved surface approximation. *J Inf Comput Sci* 11(6):2011–2018
13. Masuda H, He J (2015) TIN generation and point-cloud compression for vehicle-based mobile mapping systems. *Adv Eng Inform* 29(4):841–850
14. Panholzer H, Prokop A (2013) Wedge-filtering of geomorphologic terrestrial laser scan data. *Sensors* 13(2):2579–2594
15. Axelsson P (1999) Processing of laser scanner data—algorithms and applications. *ISPRS J Photogrammetry Remote Sens* 54(2–3):138–147
16. Wei Z, Ma H, Chen X, Liu L (2017) An improved progressive triangulation algorithm for vehicle-borne laser point cloud. *Int Arch Photogrammetry Remote Sens Spat Inf Sci* 42(2W7):929–933
17. Vosselman G, Coenen M, Rottensteiner F (2017) Contextual segment-based classification of airborne laser scanner data. *ISPRS J Photogrammetry Remote Sens* 128:354–371
18. Yilmaz M, Uysal M (2016) Comparison of data reduction algorithms for LiDAR-derived digital terrain model generalisation. *Area* 48(4):521–532
19. Dore C, Murphy M (2017) Current state of the art historic building information modelling. *Int Arch Photogrammetry Remote Sens Spat Inf Sci* 42:185–192